RFC # 724
NIC #37435

12 May 1977

Proposed Official Standard for the Format of ARPA Network Messages

by

Ken Pogran, MIT-LCS/CSR (Pogran at MIT-Multics)
John Vittal, BBN (Vittal at BBN-TENEXA)
Dave Crocker, RAND-ISD (DCrocker at Rand-Unix)
Austin Henderson, BBN (Henderson at BBN-TENEXA)

PREFACE

ARPA's Committee on Computer-Aided Human Communication (CAHCOM) wishes to promulgate an official standard for the format of ARPA Network mail headers which will adequately meet the needs of the various message service subsystems on the Network today. The authors of this RFC constitute the CAHCOM subcommittee charged with the task of developing this new standard; this document presents our current thoughts on the matter and a specific proposal.

This document is organized as follows: First, we present a history, of the development of what has become known as the ARPA Network "mail" or "message" service, and the issues which we feel are most pressing -- problems for which solutions are lacking today, inhibiting the further development of message subsystems. We then present the specification for the new ARPA Network Message Header standard. This is followed by a References section.

Essentially, we propose a revision to Request for Comments (RFC) 561, "Standardizing Network Mail Headers", and RFC 680, "Message Transmission Protocol". This revision removes and compacts portions of the previous syntax and adds several features to network address specification. In particular, we focus on people and not mailboxes as recipients and allow reference to stored address lists. We expect this syntax to provide sufficient capabilities to meet most users' immediate needs and, therefore, give developers enough breathing room to produce a new mail transmission protocol "properly". We believe that there is enough of a consensus in the Network community in favor of such a standard syntax to make possible its adoption at this time.

We would like to make clear the status of this proposed standard: The CAHCOM Steering Committee has replaced the Message Service Committee as the ARPANET standards-setting organization in the area of message services. It is expected that the proposal of this CAHCOM subcommittee, when in its final form, will be adopted as an ARPANET standard by CAHCOM. In the interests of making this standard the best possible one, we are distributing this proposal as an RFC.

Please send any comments and criticisms to any of the authors of this RFC by 15 June 1977. It is planned that the standard will be officially adopted by 1 September 1977, with hosts expected to accept its syntax by 1 January 1978.

CONTENTS

I. PROBLEMS WITH ARPANET
MESSAGE STANDARDS

A. Background and History
B. Issues and Conclusions
C. Message Parts
D. Adoption of the Standard

II. STANDARD FOR THE FORMAT
OF ARPA NETWORK MESSAGES

A. Framework
B. Syntax
C. Semantics
D. Examples

III. REFERENCES

APPENDIX

A. Alphabetical Listing of Syntax Rules

I. PROBLEMS WITH ARPANET MESSAGE STANDARDS

A. BACKGROUND AND HISTORY

Today's ARPA Network "mail" or "message" service uses, for its delivery mechanism, two special commands of the File Transfer Protocol. Viewed from within the structure of FTP, the entire message, both header and text, is data for the FTP MAIL and MLFL commands. This facility was added to the File Transfer Protocol as an afterthought; it was an interim solution to be used only until a separate mail transmission protocol was specified. Several versions of such a protocol have been proposed, but none has yet received general acceptance. Meanwhile, attempts have been made to improve upon the original interim facility.

As message service subsystems on various host systems (especially TENEX) developed to the point where rudimentary parsing of incoming messages was being done, it became clear that it would be desirable to standardize the format and content of the headers of messages transmitted between hosts using these FTP commands. To this end, an ad hoc committee wrote RFC 561, which suggested a standard message header format. The committee was unofficial, so it could not legislate a standard, it could only recommend. However, the standard it suggested adequately met an urgent need, and was generally adopted.

Several salient points should be noted:

1. RFC 561 defined the concept of a message header, and specified the syntax which delimited it from the actual text of a message;

2. It proposed a standard format for the most obvious and most urgently-needed header items: "From:", "Date:", and "Subject:";

3. It proposed that a general standard syntax be used for all other header items;

4. RFC 561 is still, today, an unofficial standard, adhered to by most because of its utility;

5. Its syntax was designed to allow humans to read the text easily, without the aid of special message processing systems.

As message services grew in sophistication, the need for specific header items in RFC 561's "miscellaneous" category grew: "To:" and "cc:", especially, were generated and recognized by several different message services. However, there was no specific standard for the syntax of the contents of these items. The message service subsystems on TENEX developed a particular format for these items; since more messages originated from the TENEX hosts on the Network than from any other type of host system, the TENEX format for these fields soon became a de facto standard. Message service subsystems on TENEX began to parse these fields, expecting them to be in the TENEX-generated format. Message service subsystems on other hosts -- Multics, for example -- began to dabble with other formats for these fields, since there was no standard for them, only to receive complaints from users of TENEX message service subsystems that their "non-standard" message headers could not be parsed according to the (de facto) "standard" syntax.

Recognizing that the time had come to make an attempt to standardize the additional header fields that had come into use since RFC 561 was published, ARPA's Message Service Committee chartered a small group in 1975 to develop a revised version of RFC 561 which would define the syntax of these additional message header fields. Several things should be noted about this small group of people: first, they were TENEX-oriented; when the functionality of the message header items they desired was matched by the functionality of an already-existing message header item of the TENEX message subsystems, they adopted the syntax used by the TENEX message subsystems. Second, they based additional header items not already found on TENEX message subsystems on the deliberations of the Message Service Committee. Third, they were not familiar with the procedure for publication of a document as a Network RFC.

The document which this group produced, labelled RFC 680, "Message Transmission Protocol", received only limited distribution. Matters were further confused because its title was misleading, since it was not a protocol for the transmission of messages between ARPA Network hosts, but rather a standard for the format of messages transmitted via the standard File Transfer Protocol. Some, including the Message Service Committee, believed that RFC 680 became a Network Standard. This was not strictly true, because it never received proper distribution, and it had never been "officially blessed" by anyone, to turn it from a request for comments into an accepted official ARPA Network standard document. Reflecting this confusion over the status of the document are the facts that the document DOES currently reside in the "official" ARPANET Protocol Handbook, and most users and message system implementors remain unaware that this is so.

For all its shortcomings, RFC 680 has performed a needed service, just as did RFC 561 before it. It defined additional message header items at a time when this needed to be done. Unfortunately, since the group had not sought ideas and input from others, the specification did not adequately respond to a sufficient set of community needs. In addition, the manner in which the document was promulgated -- or not promulgated -- left a great deal to be desired. Implementators of message-processing subsystems who had not received RFC 680 proceeded to go their own ways, feeling justified in doing so, while those who accepted RFC 680 as a standard felt justified in complaining to -- and about -- those whom they considered to be maverick implementors of idiosyncratic message service subsystems.

Perhaps because of the ad-hoc nature of the interim mail facility, users have not, until recently, attempted to push the system to the limits of their imagination. Presently, however, several different sites are using the "interim" mail facility for more than it was designed and in ways which are incompatible both with each other and with the original intent of the facility. Mail subsystem implementors are increasingly being asked to provide for the handling of mail from idiosyncratic hosts. Also, it has become clear that there are a few very specific features, too useful to ignore, which cannot reasonably be specified within the syntax of RFC 680.

B. ISSUES AND CONCLUSIONS

At first glance, it would seem that a resolution of today's somewhat chaotic situation could best be obtained by immediately junking the existing "interim" mail facility, and adopting a true mail transmission protocol. We strongly believe that this would be ill-advised at this time, for we feel that there is no general understanding within the Network community today of how to specify and implement a full and adequate mail transmission protocol. However, we are convinced that there is, finally, a strong commitment within the Network community to attack this problem (which there was not at the time the "interim" mail transmission facility was specified and developed).

The frontal attacks on the mail protocol problem have, so far, resulted in at least two suggestions for a mail transmission protocol. Why should not one of these protocols be adopted immediately? We feel that, in general, there has been a tendency for experimental Network software to be prematurely treated as though it were adequately designed and fully operational. Typically, the system or protocol proposed is so much better than what was previously available that its experimental nature is disregarded, and it is pressed into service before it has had a

chance to properly develop and mature. We are very concerned that this phenomenon not afflict the Network mail system any more than it already has.

While it is true that there are several sites in the ARPA Community which have mail systems that understand the syntax specified in RFC's 561 and 680, in addition to some of the "non-standard" syntax provided by the mail generating programs at several other sites, most mail systems do not parse much of the contents of received messages. A consideration of the syntax specified here is that messages which are sent to people should be easily read by people. Parsers which can turn an ugly, syntactically expedient form into something which is easy to read are the exception, rather than the rule in today's message systems. Also, the modifications to the existing "non-standard" syntax should be kept to a minimum, enhancing the probability that the requirement of small perturbations to existing software will be accepted.

With this syntax, we introduce mechanisms so that:

1. Users of mail systems can have multiple mailboxes, either on one machine or multiple machines, all of which are treated identically; the default mailbox for a user is not necessarily associated (directly) with his login name.

2. Mail for a person can be sent to other than a single, default mailbox.

3. Named groups may consist of both individuals and (possibly) other named groups (i.e., nesting within groups is permitted).

4. Address lists may contain references to other, stored, lists. The complete path with which one can retrieve the stored list may be specified in order to allow either manual or automatic retrieval of the stored list.

5. Address lists may contain references to addresses which are not accessible through the standard ARPANET message system. For example, U.S. Postal system addresses can be specified. Such addresses are, of course, expected to be ignored by the ARPANET system, although individual sites may provide services for using the information (e.g., automatically sending a copy of the message to a line printer, in preparation for transmission through the Postal system).

6. Parenthetical remarks, or comments, can be included and syntactically recognized as such within some header items.

7. Received messages are capable of being read by humans without a program having to parse the message (or parts of it) before presenting the message to the user; however there is sufficient formal syntax to enable a parsing program to modify the appearance and content of material presented to users. Although message-display software may exercise considerable control over message appearance, the degree to which a message's actual format is PLEASANT for humans to read is entirely the responsibility of the message creation program.

No mechanism for authentication is provided, since the Network provides no mechanisms for enforcing mail security. The syntax does provide for one aspect of "correctness": a distinction is made between an address which is claimed to be a valid network address and one which is simply free text, included for the convenience of the human participants.

C. MESSAGE PARTS

Some confusion has existed over the roles played by different message parts. Einar Stefferud has suggested using the perspective of envelope, letter head, and letter content. The presence of structured portions in messages additionally requires reference to "headers".

In computer-based message systems, human users do not generally encounter "envelopes", which are often constructed automatically, to be used by the participating system(s) to deliver the message. For example on TENEX, the envelope is the name of the file containing a message awaiting transmission. For FTP servers, it is the data portion of the MAIL or MLFL command line. Some systems attach "envelope-like" information to the message header, such as time- stamp and originating host name.

In paper-based communications, headers occur both before (e.g., "To:" and "From:" and after (e.g., "cc:" and "enclosure:") the body of the message. Within this standard, all headers occur before the body of the message, although local message display programs may choose to alter that ordering.

Wayne Hathaway has pointed out that ARPANET message format does not support specification of letterheads, since these are a type of organizational public relations symbol. Some idiosyncrasies are supported, however, by way of choosing special field names.

In general, it is important to realize that the header portion of a message plays several roles during the life of a

message, variously participating in each of the three functions suggested by Stefferud.

D. ADOPTION OF THE STANDARD

During the early phases of specifying this standard, a great deal of concern was expressed over the problems which may be experienced during the transition from the current standard to this new one. We feel that the true problem is the lack of realization that THERE IS NO CURRENT OFFICIAL STANDARD. Enough systems have enough overlapping behaviors to allow the current mail environment to function, but this in no way constitutes a standard.

In fact, we strongly believe that the new requirements imposed by the proposed standard involve less complexity than the ambiguities resulting from the current variations in system behaviors.

II. STANDARD FOR THE FORMAT OF ARPA NETWORK MESSAGES

This standard supercedes the informal standards specified in ARPANET Request for Comments numbers 561, "Standardizing Network Mail Headers", and 680, "Message Transmission Protocol". In this document, a general framework is described. The formal syntax is then specified, followed by a discussion of the semantics. Finally, a number of examples are given.

This specification is intended strictly as a definition of what is to be passed between hosts on the ARPANET. It is NOT intended to dictate either features which systems on the Network are expected to support, or user interfaces to message creating or reading programs.

A distinction should be made between what the specification requires and what it allows. Certain equivalences are defined, such as between a space character <space> and an end-of-line character <crlf>, which both facilitate the formal specification and indicate what the OFFICIAL semantics are for messages. Particular implementations may wish to preserve further distinctions which the specification does not require.

A. FRAMEWORK

Since there are many message systems which exist outside the ARPANET environment, as well as those within it, it may be useful to consider the general framework, and resulting capabilities and limitations, of this standard.

Messages are expected to consist of lines of text. No special provisions are made, at this time, for encoding drawings, facimile, speech, or structured text.

No significant consideration has been given to questions of data compression or transmission/storage efficiency. The standard, in fact, tends to be very free with the number of bits consumed. For example, field names are specified as free text, rather than special terse codes.

A general "memo" framework is used. That is, a message consists of some information, in a rigid format, followed by the main part of the message, which is text and whose format is not

specified in this document. The syntax of several fields of the rigidly-formated ("header") section is defined in this specification; some of the header fields must be included in all messages. In addition to the fields specified in this document, it is expected that other fields will gain common use. User-defined header fields allow systems to extend their functionality while maintaining a uniform framework. Our approach is similar to that of the TELNET protocol, in that we are defining a basic standard which includes a mechanism for (optionally) extending itself. The authors of this document will regulate the publishing of specifications for these extensions.

Such a framework severely constrains document "tone" and appearance and is primarily useful for most intra-organization communications and relatively structured inter-organization communication. A more robust environment might allow for multi-font, multi-color, multi-dimension encoding of information. A less robust environment, as is present in most single-machine message systems, would more severely constrain the ability to add fields and the decision to include specific fields. Relative to paper-based communication, it is interesting to note that the RECEIVER of a message can exercise an extraordinary amount of control over the message's appearance. The amount of actual control available to message receivers is contingent upon the capabilties of their individual message systems.

B. SYNTAX

This syntax is given in four parts. The first part describes a base-level lexical analyzer which feeds the higher-level parser described in the succeeding sections. The second part gives a general syntax for messages and standard header fields. The third part specifies the syntax of addresses. A final section specifies some general syntax which supports the other sections.

1. LEXICAL ANALYSIS OF MESSAGES

a. General Description

A message consists of headers and, optionally, a body (i.e. the <message-text>). The <message-text> part is just a sequence of ASCII characters; it is separated from the headers by a null line (i.e., a line with nothing preceding the <crlf>).

1) Folding and unfolding of headers

Each header item can be viewed as a single, logical, long line of ASCII characters. For convenience, this conceptual entity can be split into a multiple-line representation (i.e., "folded"). The general rule is that wherever there can be <linear-white-space> characters, you can instead insert a <crlf> immediately followed by AT LEAST one <linear- white-space> character. Thus, the single line

```
To:  "Joe Dokes & J. Harvey" <ddd at Host>, JJV at BBN
```

can be represented as

```
To:  "Joe Dokes & J. Harvey" <ddd at Host>,
     JJV at BBN
```

and

```
To:  "Joe Dokes & J. Harvey"
                  <ddd at Host>,
 JJV at BBN
```

and

To: "Joe Dokes
& J. Harvey" <ddd at Host>, JJV at BBN

The process of moving from this folded multiple-line representation of a header field to its single line representation will be called "unfolding". Unfolding is accomplished by regarding <crlf> immediately followed by a <linear-white-space-char> as equivalent to the <linear-white-space-char>.

2) Structure of header fields

Once header fields have been unfolded, they may be viewed as being composed of a <field-name> followed by a ":" (colon), followed by a <field-body>. The <field-name> must be composed of printable ASCII characters (i.e., characters which have decimal values between 33 and 126) and <linear-white-space> characters. The <field-body> may composed of any ASCII characters (other than <cr> and <lf>, which have been removed by unfolding).

Certain header fields
may be interpreted according to an internal syntax which some systems may wish to parse. These fields will be referred to as structured fields. Examples include fields containing dates and addresses. Other fields, such as the subject field, are regarded simply as a single line of text.

3) Field names

To aid in the creation and reading of <field-name>s, the free insertion of <linear-white-space> characters is allowed in reasonable places. Rather than obscuring the syntax specification for <field-name> with the explicit syntax for these <linear-white-space> characters, the existence of a simple "lexical" analyzer is assumed. The analyzer reinterprets the unfolded text which comprises the <field-name> as a sequence of <atoms> separated by <linear-white-space> characters. The field name may be conveniently represented by the sequence of these atoms, separated by a single ASCII space character.

4) Field bodies

To aid in the creation and reading of structured fields, the free insertion of <linear-white-space> characters is allowed in reasonable places. Rather than obscuring the syntax specifications for these structured fields with explicit syntax for these <linear-white-space> characters, the existence of another simple "lexical" analyzer is assumed. It provides an interpretation of the unfolded text comprising the body of the field as a sequence of lexical symbols. These include

- individual special characters
- quoted strings
- comments
- atoms

The first three symbols are self-delimiting. Atoms are not; they therefore are delimited by the self-delimiting symbols and by <linear-white-space>.

So, for example, the folded body of an address field

```
":sysmail"@ Some-Host,
Muhammed(I am the greatest)Ali at WBA
```

is analyzed into the following lexical symbols and types:

":sysmail"	quoted string
@	special
Some-Host	atom
,	special
Muhammed	atom
(I am the greatest)	comment
Ali	atom
at	atom
WBA	atom

b. Formal Definition

```
<field>          ::=  <field-name> ":" <field-body>
<field-name>     ::=  <atom>
                    | <atom> <field-name>

<field-body>     ::=  <field-body-contents>
                    | <field-body-contents> <crlf>
                        <linear-white-space-char>
                        <field-body>
```

```
<field-body-contents> ::= <the TELNET ASCII characters making
                           up the <field-body>, as defined in
                           the following sections, and
                           consisting of combinations of
                           <atom>, <quoted-string>, <text-line>,
                           and <specials> tokens>

<atom>                ::=  <a sequence of one or more TELNET
                           ASCII alpha-numeric or graphics
                           characters, excluding all control
                           characters (those characters with
                           a decimal value less than 33 or
                           equal to 127) and <delimeters> >

<quoted-string>       ::=  <double quote mark ("), decimal 34>
                           <a sequence of one or more TELNET
                           ASCII characters, where two
                           adjacent quotes are treated as a
                           single quote and part of the
                           string> <">

<text-line>           ::=  <a sequence of one or more TELNET
                           ASCII characters excluding <cr>
                           and <lf> >

<message-text>        ::=  <a sequence of zero of more TELNET
                           ASCII characters>

<delimeters>          ::=  <specials> | <comment>
                         | <linear-white-space> | <crlf>

<specials>            ::=  "(" | ")" | "<" | ">"
                         | "@" | "," | ";" | ":" | <">

<comment>             ::=  "(" <TELNET ASCII characters, except
                           <crlf> > ")"

<linear-white-space>::= <linear-white-space-char>
                        | <linear-white-space-char>
                           <linear-white-space>
<linear-white-space-char>::= <space> | <horizontal-tab>

<space>               ::=  <TELNET ASCII space (decimal 32)>
<tab>                 ::=  <TELNET ASCII tab   (decimal  9)>
<cr>                  ::=  <TELNET ASCII carriage return
                           (decimal 13)>
<lf>                  ::=  <TELNET ASCII line feed (decimal 10)>
<crlf>                ::=  <TELNET ASCII carriage return/line
                           feed (decimal 13, followed by
                           decimal 10)>
```

c. Clarifications

1) Comments

Comments may appear only within <field-body>s of structured fields. A comment is any set of TELNET ASCII characters, which is not within a quoted string, and which is enclosed in matching parentheses; parentheses nest, so that if a left paren occurs in a comment string, there must also be a matching right paren.

Comments are NOT passed to the FTP server, as part of a MAIL or MLFL command, since comments are not part of the "formal" address.

2) "White space"

Remember that in structured fields, MULTIPLE LINEAR WHITE SPACE TELNET ASCII CHARACTERS (namely <tab>s and <space>s) ARE TREATED AS SINGLE SPACES AND MAY FREELY SURROUND ANY SYMBOL. In all header fields, at least one <space> is REQUIRED only at the beginning of folded lines.

Writers of mail-sending (i.e. header generating) programs should realize that there is no Network-wide definition of the effect of <tab> TELNET ASCII characters on the appearance of text at another Network host; therefore, the use of <tab>s in message headers, though permitted, is discouraged.

Note that the contents of messages are required to conform with TELNET NVT conventions (e.g. <cr> must be followed by either <lf>, making a <crlf>, or <null>, if the <cr> is to stand alone).

3) Quoted strings

Where permitted (i.e., in structured fields) quoted strings are treated as a single symbol (i.e. equivalent to an <atom> syntactically). However, if quoted strings are to be "folded" onto multiple lines, then the syntax for folding must be adhered to (See items II.B.1.a.1, above, and II.B.1.c.6, below.) Note that the official semantics do not encounter <crlf>s in quoted strings, although particular parsing programs may wish to note their presence.

4) Bracketing characters

There are two types of brackets which must be well nested:

- Parentheses are used to indicate comments.

- Angle brackets ("<" and ">") are used where there is a question of the presence of machine-usable code (e.g. deliminating mailboxes).

5) Case independence of certain specials <atom>s

It should be assumed by all mail reading programs that certain <atom>s can be represented in any combination of upper and lower case. These are:

- <field-name>s,
- "File", in a <path>,
- "at", in an <at-indicator>,
- <host-name>s,
- <day-of-week>s,
- <string-month>s, and
- <time-zone>s

For example, the <field-name>s "From", "FROM", "from", and even "FroM" should all be treated identically. Note that, at the level of this specification, case IS relevant to other <word>s and <text-line>s. Also see Section II.C.1.a.4, below.

6) Folding long lines

Each header item (field of the message) may be represented on exactly one line consisting of the name of the field and its body, and this is what the parser sees. For readability, it is recommended that the <field-body> portion of long header items be "folded" onto multiple lines of the actual header.

7) Backspace characters

Backspace TELNET ASCII characters (ASCII BS, decimal 8) may be included in <text-line> and <quoted-string> to effect overstriking; however, any use of backspaces which effects an overstrike to the left of the beginning of the <text-line> or <quoted-string> is prohibited.

2. GENERAL SYNTAX OF MESSAGES:

NOTE: The syntax indicates that items in <required-headers> must be in a specific order and precede all other header items. Header fields, in fact, are NOT required to occur in any particular order. Required header items must be unique (occur exactly once). This specification permits multiple occurrences of most optional fields. However, the interpretation of such multiple occurrences is not specified here.

```
<message>          ::=  <headers>
                      | <headers> <crlf> <message-text>

<headers>          ::=  <required-headers>
                      | <required-headers> <optional-headers>
<required-headers> ::=  <date-field> <originator>
<originator>       ::=  <mach-from-field>
                      | <mach-from-list> <sender-field>
                      | <mach-from-field> <reply-to-field>
                      | <any-from-field> <sender-field>
                           <reply-to-field>

<date-field>       ::=  "Date"       ":" <date-time>
<mach-from-field>  ::=  "From"       ":" <mach-addr-item>
<mach-from-list>   ::=  "From"       ":" <mach-addr-list>
<any-from-field>   ::=  "From"       ":" <address-list>
<sender-field>     ::=  "Sender"     ":" <host-phrase>
<reply-to-field>   ::=  "Reply-To"   ":" <mach-addr-list>

<optional-headers>::=  <optional-header-field>
                      | <optional-headers>
                           <optional-header-field>

<optional-header-field> ::= <addressee-field>
                          | <extension-field>

<addressee-field> ::=  "To"          ":" <address-list>
                      | "cc"         ":" <address-list>
                      | "bcc"        ":" <address-list>
                      | "Fcc"        ":" <path-list>

<extension-field> ::=  "In-Reply-To" ":" <reference-list>
                      | "Keywords"   ":" <phrase-list>
                      | "Message-Id" ":" <mach-host-phrase>
                      | "References" ":" <reference-list>
                      | "Subject"    ":" <text-line>
                      | "Comments"   ":" <text-line>
                      | <user-defined-field>
```

<user-defined-field> ::= <A <field> which has a <field-name>
not defined in this specification>

The following syntax for the bodies of various fields should be thought of as describing each field body as a single long string (or line). The section on Lexical Analysis (section II.B.1) indicated how such long strings can be represented on more than one line in the actual transmitted message.

3. SYNTAX OF GENERAL ADDRESSEE ITEMS

```
<mach-addr-list>  ::=   <mach-addr-item>
                      | <mach-addr-item> "," <address-list>

<address-list>    ::=   <null>
                      | <address-item>
                      | <address-item> "," <address-list>
<address-item>    ::=   <mach-addr-item>
                      | <group-name> ":" <address-list> ";"
                      | <any-name>
                      | <path>

<mach-addr-item>  ::=   <mailbox>
                      | <phrase> "<" <mailbox-list> ">"

<group-name>      ::=   <phrase>
<any-name>        ::=   <quoted-string>

<mailbox-list>    ::=   <mailbox>
                      | <mailbox> "," <mailbox-list>
<mailbox>         ::=   <host-phrase>

<path>            ::=   ":" "File" ":" <path-name>
<path-name>       ::=   <path-item>
                      | "<" <path-list> ">"
<path-list>       ::=   <path-item>
                      | <path-item> "," <path-list>
<path-item>       ::=   <host-phrase>
```

4. SUPPORTING SYNTAX

```
<reference-list>  ::=  <null>
                     | <reference-item>
                     | <reference-item> "," <reference-list>
<reference-item>  ::=  <phrase>
                     | <mach-host-phrase>

<mach-host-phrase>::=  "<" <host-phrase> ">"
<host-phrase>     ::=  <phrase> <host-indicator>
<host-indicator>  ::=  <at-indicator> <host-name>
<at-indicator>    ::=  "at" | "@"
<host-name>       ::=  <atom>
                     | <decimal host address>

<date-time>       ::=  <day> <date> <time>
<day>             ::=  <null>
                     | <day-of-week> ","
<day-of-week>     ::=  "Monday"    | "Mon"
                     | "Tuesday"   | "Tue"
                     | "Wednesday" | "Wed"
                     | "Thursday"  | "Thu"
                     | "Friday"    | "Fri"
                     | "Saturday"  | "Sat"
                     | "Sunday"    | "Sun"
<date>            ::=  <string-date>
                     | <slash-date>
<string-date>     ::=  <day-of-month> <string-month>
                                   <4-digit-year>
<slash-date>      ::=  <numeric-month> "/" <date-of-month>
                                       "/" <2-digit-year>
<numeric-month>   ::=  <one or two decimal digits>
<day-of-month>    ::=  <one or two decimal digits>
<string-month>    ::=  "January" | "Jan"
                     | "February" | "Feb"
                     | "March"    | "Mar"
                     | "April"    | "Apr"
                     | "May"
                     | "June"     | "Jun"
                     | "July"     | "Jul"
                     | "August"   | "Aug"
                     | "September"| "Sep"
                     | "October"  | "Oct"
                     | "November" | "Nov"
                     | "December" | "Dec"
<4-digit-year>    ::=  <four decimal digits>
<2-digit-year>    ::=  <two decimal digits>
<time>            ::=  <24-hour-time> "-" <time-zone>
<24-hour-time>    ::=  <hour> <minute>
<hour>            ::=  <two decimal digits>
<minute>          ::=  <two decimal digits>
```

```
<time-zone>       ::=  "GMT" | "Z"   | "GDT"
                     | "AST" | "ADT"
                     | "EST" | "EDT" | "CST" | "CDT"
                     | "MST" | "MDT" | "PST" | "PDT"
                     | "YST" | "YDT" | "HST" | "HDT"

<phrase>          ::=  <word>
                     | <word> <phrase>
<phrase-list>     ::=  <null>
                     | <phrase>
                     | <phrase> "," <phrase-list>

<word>            ::=  <atom>
                     | <quoted-string>
```

C. SEMANTICS

1. ADDRESS FIELDS

a. General

1) <path>s are used to refer to a location, on the ARPANET containing a stored address list. The <phrase> should contain text which the referenced host can resolve to a file. This standard is not a protocol and so does not prescribe HOW data is to be retrieved from the file. However, the following requirements are made:

- the file must be accessible through the local operating system interface (if it exists), given adequate user access rights; and

- if a host has an FTP server and a user is able to retrieve any files from the host using that server, then the file must be accessible through FTP, using DEFAULT transfer settings, given adequate user access rights.

It is intended that this mechanism will allow programs to retrieve such lists automatically.

The interpretation of a <path> follows. This is not intended to imply any particular implementation scheme, but is included to aid in understanding the notion of <path>s:

- The contents of the file indicated by a <path-name> is treated as an <address-list> and is inserted as an <address-item> in the position of the <path-name> item in the syntax. That is, the TELNET ASCII character string of the <path-name> or, if present, the <path-list> containing it, is replaced by the contents of the file to which the <path-name> refers. Therefore, the contents of the file indicated by a <path-name> must be syntactically self-contained and must adhere to the full syntax prescribed herein for <address-list>.

- <Path-item>s of a <path-list> are alternates and the contents of ONLY ONE of them is to be included in the resultant address list.

2) The <phrase> part of a <mailbox> is understood to be whatever the receiving FTP Server allows (for example,

TENEX systems do not now understand addresses of the form "P. D. Q. Bach", but another system might).

Note that a <mailbox> is a conceptual entity which does not necessarily pertain to file storage. For example, some sites may choose to print mail on their line printer and deliver the output to the addressee's desk.

A user may have several mailboxes. The use of the second alternative of <mach-addr-item> (<phrase> "<" <mailbox-list> ">") indicates that a copy of the message is to be sent to EACH mailbox named.

3) <any-name> may contain any sequence of "words". This sequence of words, used as an <address-item>, is used to facilitate reference to non-standard (e.g. non-Network) addresses. Such an address might be one which is acceptable to the U.S. Postal Service.

4) The <host-name> in a <host-phrase> must be THE official name of a Network host, or else a decimal number indicating the Network address for that host. The USE OF NUMBERS IS STRONGLY DISCOURAGED and is permitted only due to the occasional necessity of bypassing local host-name tables.

The <phrase> in a <host-phrase> is intended to be meaningful only to the indicated host. To all other hosts, the <phrase> is treated as a literal string. No case transformations should be (automatically) performed on the <phrase>. The <phrase> is passed to the local host's mail sending program; it is the responsibility of the destination host's mail receiving (distribution) program to perform case mapping on this <phrase>, if required, to deliver the mail.

b. Originator Fields

WARNING: The standard allows only a subset of the combinations possible with the From, Sender, and Reply-to fields. The limitation is intentional; the permitted alternatives have been carefully chosen and are adequate for the purposes of this standard.

1) From:

This field contains the identity of the person(s) who wished this message to be sent. The message-creation process should default this field to be a single machine address, indicating the user entering the message; if and only if this is done, the "Sender:" field need not be present.

2) Sender:

This field contains the identity of the person who sends the message. It need not be present in the header of the message if it is the SAME as the "From:" field.

The <sender-field-body> includes a <phrase> which must correspond to a user, rather than a standard <address-item>, to indicate the expectation that the field will refer to the PERSON responsible for sending the mail and not simply include the name of a mailbox, from which the mail was sent. For example in the case of a shared login name, the name, by itself, would not be adequate. The <phrase> (user) is a system entity, not a generalized person reference.

3) Reply-to:

This field provides a general mechanism for indicating any mailbox(es) to which responses are to be sent. Three different uses for this feature can be distinguished. In the first case, the author(s) may not have regular machine-based mailboxes and therefore wish to indicate an alternate machine address. In the second case, an author may wish additional persons to be made aware of, or responsible for, responses; responders should send their replies to the "Reply-to:" mailbox(es). More interesting is a case such as text-message teleconferencing in which an automatic distribution facility is provided and a user submitting an "entry" for distribution only needs to send their message to the mailbox(es) indicated in the "Reply-to:" field.

If there is no <reply-to-field>, then the <from-field> MUST contain AT LEAST ONE machine address. In all cases when used and even if a <sender> field is present, the Reply-to field must contain at least one machine address.

NOTE: For systems which automatically generate address lists for replies to messages, the following requirements are made:

- The receiver, when replying to a message, must NEVER automatically include the <sender-field-body> in the reply's address list

- If the <reply-to-field> exists, then the reply should go ONLY to the <reply-to-field-body> addressees.

(Extensive examples are provided in Section II.D.) This recommendation is intended only for <originator-field>s and in no way is intended to reflect that replies should not be sent, also, to the other recipients of this message. It is up to the respective mail handling programs as to what additional facilities will be provided.

c. Receiver Fields

1) To:

This field contains the identity of the primary recipients of the message.

2) cc:

This field contains the identity of the secondary recipients of the message.

3) Bcc:

This field contains the identity of additional recipients of the message who are to remain hidden from the primary and secondary recipients. Some systems may choose to include the text of the "Bcc:" field only in the author(s)'s copy, while others may include it in the text sent to all those indicated in the "Bcc:" list.

4) Fcc:

This field contains the identity of any message files in which copies of this message are being placed by the originator. Note that the presence of this field does NOT guarantee long-term availability of the message in any of the indicated files.

2. REFERENCE SPECIFICATION FIELDS

a. Message-Id:

This field contains a unique identifier (the <phrase>) to refer to this version of this message. The uniqueness of the message identifier is guaranteed by each host. This identifier is intended to be machine readable, and not necessarily meaningful to humans. A message-id pertains to exactly one instantiation of a particular message; subsequent revisions to the message should receive new message-id's.

b. In-Reply-To:

The contents of this field identify previous correspondence which this message answers. If message identifiers are used in this field, they should be enclosed in angle brackets (<>).

c. References:

The contents of this field identify other correspondence which this message references. If message identifiers are used, they should be enclosed in angle brackets (<>).

d. Keywords:

This field contains keywords or phrases, separated by commas.

3. OTHER FIELDS AND SYNTACTIC ITEMS

a. Subject:

The "subject:" field is intended to provide as much information as necessary to adequately summarize or indicate the nature of the message.

b. Comments:

Permits adding text comments onto the message without disturbing the contents of the message's body.

4. DATES

It is recommended that, because of differing international interpretations, the <string-day> option be used instead of the <slash-day> option in the specification of a <day>.

If included, <day-of-week> must be the day implied by the <date> specification.

<Time-zones> allow reference to Greenwich and to each of the zones in the United States. The zone references beginning with "A" are for Atlantic time which are one hour faster than the corresponding Eastern times. "Y" indicates Yukon time in Alaska, which is one hour slower than the corresponding Pacific times, and "H" indicates Hawaiian times, which are two hours slower.

D. EXAMPLES

1. ADDRESSES

a. Alfred E. Newman <Newman at BBN-TENEXA>
Newman@BBN-TENEXA

These two "Alfred E. Newman" examples have identical semantics, as far as the operation of the local host's mailer and the remote host's FTP server are concerned. In the first example, the "Alfred E. Newman" is ignored by the mailer, as "Newman at BBN-TENEXA" completely specifies the recipient. The second example contains no superfluous information, and, again, "Newman@BBN-TENEXA" is the intended recipient.

b. Al Newman at BBN-TENEXA

This is identical with "Al Newman<Al Newman at BBN-TENEXA>." That is, the full <phrase>, "Al Newman", is passed to the FTP server. Note that not all FTP servers accept multi-word identifiers; and some that do accept them will treat each word as a different addressee (in this case, attempting to send a copy of the message to "Al" and a copy to "Newman").

c. "George Lovell, Ted Hackle" <Shared-Mailbox at Office-1>

This form might be used to indicate that a single mailbox is shared by several users. The quoted string is ignored by the originating host's mailer, as "Shared-Mailbox at Office-1" completely specifies the destination mailbox.

d. Wilt (the Stilt) Chamberlain at NBA

The "(the Stilt)" is a comment, which is NOT included in the destination mailbox address handed to the originating system's mailer. The address is the string "Wilt Chamberlain", with exactly one space between the first and second words. (The quotation marks are not included.)

2. ADDRESS LISTS

```
Gourmets:  Pompous Person <WhoZiWhatZit at Cordon-Bleu>,
           Cooks:  Childs at WGBH, Galloping Gourmet at
                   ANT (Australian National Television);
           Wine Lovers:  Drunk at Discount-Liquors,
                         Port at Portugal;;,
Jones at SEA
```

This group list example points out the use of comments, the nesting of groups, and the mixing of addresses and groups. Note that the two consecutive semi-colons preceding "Jones at SEA" mean that Jones is NOT a member of the Gourmets group.

3. ORIGINATOR ITEMS

a. George Jones logs into his Host as "Jones". He sends mail himself.

```
From:  Jones at Host
```

or

```
From:  George Jones <Jones at Host>
```

b. George Jones logs in as Jones on his Host. His secretary, who logs in as Secy on her Host (SHost) sends mail for him. Replies to the mail should go to George, of course.

```
From:    George Jones <Jones at Host>
Sender:  Secy at SHost
```

c. George Jones logs in as Group at Host. He sends mail himself; replies should go to the Group mailbox.

```
From:  George Jones <Group at Host>
```

d. George Jones' secretary sends mail for George in his capacity as a member of Group while logged in as Secy at Host. Replies should go to Group.

```
From:    George Jones<Group at Host>
Sender:  Secy at Host
```

Note that there need not be a space between "Jones" and the "<", but adding a space enhances readability (as is the case in other examples).

e. George Jones asks his secretary (Secy at Host) to send a message for him in his capacity as Group. He wants his secretary to handle all replies.

From: George Jones <Group at Host>
Sender: Secy at Host
Reply-to: Secy at Host

f. A non-ARPANET user friend of George's, Sarah, is visting. George's secretary sends some mail to a friend of Sarah in computer-land. Replies should go to George, whose mailbox is Jones at Host.

From: Sarah Friendly
Sender: Secy at Host
Reply-to: Jones at Host

g. George is a member of a committee. He wishes to have any replies to his message go to all committee members.

From: George Jones
Sender: Jones at Host
Reply-To: Big-committee: Jones at Host,
Smith at Other-Host,
Doe at Somewhere-Else;

Note that if George had not included himself in the enumeration of Big-committee, he would not have gotten a reply; the presence of the "Reply-to:" field SUPERSEDES the sending of a reply to the person named in the "From:" field.

h. (Example of INCORRECT USE)

George desires a reply to go to his secretary; therefore his secretary leaves his mailbox address off the "From:" field, leaving only his name, which is not, itself, a mailbox address.

From: George Jones
Sender: Secy at SHost

THIS IS NOT PERMITTED. Replies are NEVER implicitly sent to the "Sender:"; George's secretary should have used the "Reply-to:" field, or the mail creating program she was using should have forced her to.

i. George's secretary sends out a message which was authored jointly by all the members of the "Big-committee".

From: Big-committee: Jones at Host,
Smith at Other-Host,
Doe at Somewhere-Else;
Sender: Secy at SHost

4. COMPLETE HEADERS

a. Minimum required:

```
Date:  26 August 1976 1429-EDT
From:  Jones at Host
```

b. Using some of the additional fields:

```
Date  26 August 1976 1430-EDT
From:  George Jones<Group at Host>
Sender: Secy at SHOST
To:    Al Newman at Mad-Host,
       Sam Irving at Other-Host
Message-id:  some string at SHOST
```

c. About as complex as you're going to get:

```
Date:          27 Aug 1976 0932-PDT
From:          Ken Davis <KDavis at Other-Host>
Sender:        KSecy at Other-Host
Reply-to:      Sam Irving at Other-Host
Subject:       Re: The Syntax in the RFC
To:            George Jones <Group at Host>,
               Al Newman at Mad-Host
cc:            Tom Softwood <Balsa at Another-Host>,
               Sam Irving at Other-Host,
               Standard Distribution:
                :File:
                  </main/davis/people/standard at Other Host,
                  "<Jones>standard.dist.3" at Tops-20-Host>
In-Reply-to: <some string at SHOST>
Message-ID: 4231.629.XYzi-What at Other-Host
Comment:       Sam is away on business. He asked me to handle
               his mail for him today.  He'll be able to
               provide a more accurate explanation tomorrow
               when he returns.
```

III. REFERENCES

--- TELNET Protocol Specification. Network Information Center No. 18639; Augmentation Research Center, Stanford Research Institute: Menlo Park, August 1973.

Bhushan, A.K. The File Transfer Protocol. ARPANET Request for Comments, No. 354, Network Information Center No. 10596; Augmentation Research Center, Stanford Research Institute: Menlo Park, July 1972.

Bhushan, A.K. Comments on the File Transfer Protocol. ARPANET Request for Comments, No. 385, Network Information Center No. 11357; Augmentation Research Center, Stanford Research Institute: Menlo Park, August 1972.

Bhushan, A.K., Pogran, K.T., Tomlinson, R.S., and White, J.E. Standardizing Network Mail Headers. ARPANET Request for Comments, No. 561, Network Information Center No. 18516; Augmentation Research Center, Stanford Research Institute: Menlo Park, September 1973.

Feinler, E.J. and Postel, J.B. ARPANET Protocol Handbook. Network Information Center No. 7104; Augmentation Research Center, Stanford Research Institute: Menlo Park, April 1976. (NTIS AD A003890).

McKenzie, A. File Transfer Protocol. ARPANET Request for Comments, No. 454, Network Information Center No. 14333; Augmentation Research Center, Stanford Research Institute: Menlo Park, February 1973.

Myer, T.H. and Henderson, D.A. Message Transmission Protocol. ARPANET Request for Comments, No. 680, Network Information Center No. 32116; Augmentation Research Center, Stanford Research Institute: Menlo Park, 1975.

Neigus, N. File Transfer Protocol. ARPANET Request for Comments, No. 542, Network Information Center No. 17759; Augmentation Research Center, Stanford Research Institute: Menlo Park, July 1973.

Postel, J.B. Revised FTP Reply Codes. ARPANET Request for Comments, No. 640, Network Information Center No. 30843; Augmentation Research Center, Stanford Research Institute: Menlo Park, June 1974.

APPENDIX

A. ALPHABETICAL LISTING OF SYNTAX RULES

```
<2-digit-year>     ::=   <two decimal digits>
<4-digit-year>     ::=   <four decimal digits>
<24-hour-time>     ::=   <hour> <minute>

<addressee-field>  ::=   "To"          ":" <address-list>
                       | "cc"          ":" <address-list>
                       | "bcc"         ":" <address-list>
                       | "Fcc"         ":" <path-list>
<address-item>     ::=   <mach-addr-item>
                       | <group-name> ":" <address-list> ";"
                       | <any-name>
                       | <path>
<address-list>     ::=   <null> | <address-item>
                       | <address-item> "," <address-list>

<any-from-field>   ::=   "From"        ":" <address-list>

<any-name>         ::=   <quoted-string>

<at-indicator>     ::=   "at" | "@"

<atom>             ::=   <a sequence of one or more TELNET ASCII
                          alpha-numeric or graphics characters,
                          excluding all control characters
                          (those characters with a decimal value
                          less than 33 or equal to 127) and
                          <delimeters> >

<comment>          ::=   "(" <TELNET ASCII characters, except
                          <crlf> > ")"

<cr>               ::=   <TELNET ASCII carriage return (decimal 13)>
<crlf>             ::=   <TELNET ASCII carriage return/line feed
                          (decimal 13, followed by decimal 10)>

<date>             ::=   <string-date> | <slash-date>
<date-field>       ::=   "Date"        ":" <date-time>
<date-time>        ::=   <day> <date> <time>
<day>              ::=   <null> | <day-of-week> ","
<day-of-month>     ::=   <one or two decimal digits>
<day-of-week>      ::=   "Monday"    | "Mon"
                       | "Tuesday"   | "Tue"
                       | "Wednesday" | "Wed"
                       | "Thursday"  | "Thu"
```

```
                        | "Friday"    | "Fri"
                        | "Saturday"  | "Sat"
                        | "Sunday"    | "Sun"

<delimeter>         ::=   <specials> | <comment>
                        | <linear-white-space> | <crlf>

<field>             ::=   <field-name> ":" <field-body>
<field-body>        ::=   <field-body-contents>
                        | <field-body-contents> <crlf>
                             <linear-white-space-CHAR> <field-body>
<field-body-contents> ::= <the TELNET ASCII characters making up
                             the field body, as defined in the
                             following sections and consisting of
                             combinations of <atom>, <quoted-
                             string>, <text-line>, and <specials>
                             tokens>

<field-name>        ::=   <atom> | <atom> <field-name>

<group-name>        ::=   <phrase>

<headers>           ::=   <required-headers>
                        | <required-headers> <optional-headers>

<host-indicator>    ::=   <at-indicator> <host-name>
<host-name>         ::=   <atom>  | <decimal host address>
<host-phrase>       ::=   <phrase> <host-indicator>

<hour>              ::=   <two decimal digits>

<lf>                ::=   <TELNET ASCII line feed (decimal 10)>
<linear-white-space>::= <linear-white-space-char>
                        | <linear-white-space-char>
                             <linear-white-space>
<linear-white-space-char>::=  <space> | <horizontal-tab>

<mach-addr-item>    ::=   <mailbox> | <phrase> "<" <mailbox-list> ">"
<mach-addr-list>    ::=   <mach-addr-item>
                        | <mach-addr-item> "," <address-list>

<mach-from-field>   ::=   "From"          ":" <mach-addr-item>
<mach-from-list>    ::=   "From"          ":" <mach-addr-list>

<mach-host-phrase>::=   "<" <host-phrase> ">"

<mailbox>           ::=   <host-phrase>
<mailbox-list>      ::=   <mailbox> | <mailbox> "," <mailbox-list>

<message>           ::=   <headers>
                        | <headers> <crlf> <message-text>
```

```
<message-text>     ::=   <a sequence of zero of more TELNET ASCII
                            characters>

<minute>           ::=   <two decimal digits>

<numeric-month>    ::=   <one or two decimal digits>

<optional-headers>::=    <optional-header-field>
                       | <optional-headers> <optional-header-field>
<optional-header-field> ::= <addressee-field> | <extension-field>

<originator>       ::=   <mach-from-field>
                       | <mach-from-list> <sender-field>
                       | <mach-from-field> <reply-to-field>
                       | <any-from-field> <sender-field>
                            <reply-to-field>

<path>             ::=   ":" "File" ":" <path-name>
<path-item>        ::=   <host-phrase>
<path-list>        ::=   <path-item> | <path-item> "," <path-list>
<path-name>        ::=   <path-item> | "<" <path-list> ">"

<phrase>           ::=   <word> | <word> <phrase>
<phrase-list>      ::=   <null> | <phrase>
                       | <phrase> "," <phrase-list>

<reference-item>   ::=   <phrase> | <mach-host-phrase>
<reference-list>   ::=   <null> | <reference-item>
                       | <reference-item> "," <reference-list>

<quoted-string>    ::=   <double quote mark ("), decimal 34>
                            <a sequence of one or more TELNET
                            ASCII characters, where two adjacent
                            quotes are treated as a single quote
                            and part of the string> <">

<reply-to-field>   ::=   "Reply-To"    ":" <mach-addr-list>

<required-headers> ::=   <date-field> <originator>

<sender-field>     ::=   "Sender"      ":" <host-phrase>

<slash-date>       ::=   <numeric-month> "/" <date-of-month>
                                         "/" <2-digit-year>
<space>            ::=   <TELNET ASCII space (decimal 32)>

<specials>         ::=   "(" | ")" | "<" | ">"
                       | "@" | "," | ";" | ":" | <">

<string-date>      ::=   <day-of-month> <string-month>
<string-month>     ::=   "January"  | "Jan" | "February" | "Feb"
```

```
                        | "March"    | "Mar" | "April"    | "Apr"
                        | "May"                | "June"     | "Jun"
                        | "July"     | "Jul" | "August"   | "Aug"
                        | "September"| "Sep" | "October"  | "Oct"
                        | "November" | "Nov" | "December" | "Dec"

<tab>                ::=   <TELNET ASCII tab   (decimal  9)>

<text-line>           ::=   <a sequence of one or more TELNET ASCII
                              characters excluding <cr> and <lf> >

<time>               ::=   <24-hour-time> "-" <time-zone>
<time-zone>          ::=   "GMT" | "Z"   | "GDT" | "AST" | "ADT
                         | "EST" | "EDT" | "CST" | "CDT"
                         | "MST" | "MDT" | "PST" | "PDT"
                         | "YST" | "YDT" | "HST" | "HDT"

<user-defined-field> ::= <A <field> which has a <field-name> not
                              defined in this specification>

<word>               ::=   <atom> | <quoted-string>
```